1213.43347X00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): NAKAMURA, et al

Serial No.:

Filed: December 17, 2003

Title: DISTRIBUTED FILE SYSTEM

Group:

LETTER CLAIMING RIGHT OF PRIORITY

Mail Stop Patent Application
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

December 17, 2003

Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s) the right of priority based on Japanese Patent Application No.(s) 2003-063569 filed March 10, 2003.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

Carl I. Brundidge
Registration No. 29,621

CIB/nac
Attachment
(703) 312-6600

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　３月１０日

出願番号
Application Number: 特願２００３－０６３５６９

[ST. 10/C]: ［ＪＰ２００３－０６３５６９］

出願人
Applicant(s): 株式会社日立製作所

２００３年　９月２６日

特許庁長官
Commissioner,
Japan Patent Office 今井康夫

出証番号　出証特２００３－３０７９３９７

【書類名】　特許願
【整理番号】　GM0303014
【提出日】　平成15年 3月10日
【あて先】　特許庁長官殿
【国際特許分類】　G16F 12/00
【発明者】
　【住所又は居所】　東京都国分寺市東恋ヶ窪一丁目２８０番地　株式会社日立製作所　中央研究所内
　【氏名】　中村　隆喜
【発明者】
　【住所又は居所】　東京都品川区東品川四丁目１２番７号　日立ソフトウエアエンジニアリング株式会社内
　【氏名】　小山　和成
【特許出願人】
　【識別番号】　000005108
　【氏名又は名称】　株式会社日立製作所
【代理人】
　【識別番号】　100075513
　【弁理士】
　【氏名又は名称】　後藤　政喜
【選任した代理人】
　【識別番号】　100084537
　【弁理士】
　【氏名又は名称】　松田　嘉夫
【選任した代理人】
　【識別番号】　100114236
　【弁理士】
　【氏名又は名称】　藤井　正弘

【手数料の表示】

【予納台帳番号】　019839

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　0110326

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　分散ファイルシステム

【特許請求の範囲】

【請求項１】

ファイルを保持するストレージと、前記ストレージに対してファイルの操作を行う複数のクライアントと、前記クライアントによるファイルの書き込み及び読み出しの操作の権限をトークンによって管理するサーバと、がネットワークによって接続されて構成される分散ファイルシステムにおいて、

前記サーバは、前記ファイルの書込操作権限を保持しているクライアントに対して該書込操作権限に関するトークンの返却を要求するトークン返却要求を送信するトークン返却要求手段を備え、

前記トークン返却要求手段は、該ファイルに関するトークンを要求しているクライアントの情報と、該クライアントが要求しているトークンの内容を示す情報とを含めてトークン返却要求を送信することを特徴とする分散ファイルシステム。

【請求項２】

前記クライアントは、前記ストレージから読み出したファイルのデータを保持する記憶部と、

前記トークン返却要求を受信すると、該ファイルに関するトークンを前記サーバに要求しているクライアントに対して、前記記憶部に保持している該トークンに関するファイルを送信するデータ出力手段と、を備えることを特徴とする請求項１に記載の分散ファイルシステム。

【請求項３】

前記クライアントから前記トークンを前記サーバに要求しているクライアントに対して送信される該トークンに関するファイルは、前記ストレージに最新情報が反映されていないものであることを特徴とする請求項２に記載の分散ファイルシステム。

【請求項４】

前記トークンには、前記ファイルの領域に関連づけられており、

前記データ出力手段は、

前記ファイルのうち、前記トークンによって関連づけられた領域のデータを、前記トークンを前記サーバに要求しているクライアントに対して送信し、

前記ファイルのうち、前記トークンによって関連づけられていない領域のデータを、前記ストレージに書き込むことによって、前記ストレージに対する同期処理を行うことを特徴とする請求項２又は３に記載の分散ファイルシステム。

【請求項５】

前記データ出力手段は、前記保持しているファイルを前記トークンを前記サーバに要求しているクライアントに対して送信するか、前記ストレージに書き込んで、前記ストレージに対する同期処理を行うかを、前記ネットワークと前記ストレージとの入出力性能及び／又は前記トークンを前記サーバに要求しているクライアントに対して送信されるファイルのデータ量に基づいて決定することを特徴とする請求項２から４のいずれか一つに記載の分散ファイルシステム。

【請求項６】

ファイルを保持するストレージと、前記ストレージに対してファイルの操作を行う複数のクライアントと、トークンによって前記クライアントによるファイルの書き込み及び読み出しの操作の権限を管理するサーバと、がネットワークによって接続されて構成される分散ファイルシステムに用いられるファイル送受信方法であって、

前記クライアントが、前記サーバに対して、ファイルの操作権限に関するトークンを要求し、

前記サーバは、前記ファイルの書込操作権限を保持しているクライアントに対して送信される、該書込操作権限に関するトークンの返却を要求するトークン返却要求に、該ファイルに関するトークンを要求しているクライアントの情報と、該クライアントが要求しているトークンの内容を示す情報とを含めて送信することを特徴とするファイル送受信方法。

【請求項７】

前記クライアントは、前記トークン返却要求を受信すると、該ファイルに関するトークンを要求しているクライアントに対して、前記記憶部に保持している該

トークンに関するファイルを送信することを特徴とする請求項６に記載のファイル送受信方法。

【請求項８】

前記クライアントから前記トークンを前記サーバに要求しているクライアントに対して送信される該トークンに関するファイルは、前記ストレージに最新情報が反映されていないものであることを特徴とする請求項７に記載のファイル送受信方法。

【請求項９】

前記トークンには、前記ファイルの領域に関連づけられており、

前記クライアントは、

前記ファイルのうち、前記トークンによって関連づけられた領域のデータを、前記トークンを前記サーバに要求しているクライアントに対して送信し、

前記ファイルのうち、前記トークンによって関連づけられていない領域のデータを、前記ストレージに書き込むことによって、前記ストレージに対する同期処理を行うことを特徴とする請求項７又は８に記載のファイル送受信方法。

【請求項１０】

前記クライアントは、前記保持しているファイルを前記トークンを前記サーバに要求しているクライアントに対して送信するか、前記ストレージに書き込んで、前記ストレージに対する同期処理を行うかを、前記ネットワークと前記ストレージとの入出力性能及び／又は前記トークンを前記サーバに要求しているクライアントに対して送信されるファイルのデータ量に基づいて決定することを特徴とする請求項７から９のいずれか一つに記載のファイル送受信方法。

【請求項１１】

ファイルを保持するストレージと、前記ストレージに対してファイルの操作を行う複数のクライアント装置と、前記クライアント装置によるファイルの書き込み及び読み出しの操作の権限をトークンによって管理するサーバと、がネットワークによって接続されて構成される分散ファイルシステムに用いられるクライアント装置であって、

前記ストレージから読み出したファイルのデータを保持する記憶部と、

前記サーバから送信された、前記ファイルの書込操作権限に関するトークンの返却の要求を受信すると、該ファイルに関するトークンを要求しているクライアント装置に対して、前記記憶部に保持している該トークンに関するファイルを送信するデータ出力手段と、を備えることを特徴とするクライアント装置。

【請求項１２】

前記トークンを前記サーバに要求しているクライアント装置に対して送信される該トークンに関するファイルは、前記ストレージに最新情報が反映されていないものであることを特徴とする請求項１１に記載のクライアント装置。

【請求項１３】

前記データ出力手段は、

前記ファイルのうち、前記トークンによって関連づけられた領域のデータを、前記トークンを前記サーバに要求しているクライアントに対して送信し、

前記ファイルのうち、前記トークンによって関連づけられていない領域のデータを、前記ストレージに書き込むことによって、前記ストレージに対する同期処理を行うことを特徴とすることを請求項１１又は１２に記載のクライアント装置。

【請求項１４】

前記データ出力手段は、前記保持しているファイルを前記トークンを前記サーバに要求しているクライアントに対して送信するか、前記ストレージに書き込んで、前記ストレージに対する同期処理を行うかを、前記ネットワークと前記ストレージとの入出力性能及び／又は前記トークンを前記サーバに要求しているクライアントに対して送信されるファイルのデータ量に基づいて決定することを特徴とする請求項１１から１３のいずれか一つに記載のクライアント装置。

【請求項１５】

ストレージとネットワークによって接続されたクライアントによるファイルの書き込み及び読み出しの操作の権限のトークンによる管理をサーバ装置に実行させるプログラムであって、

前記サーバ装置を、前記ファイルの書込操作権限を保持しているクライアントに対して該書込操作権限に関するトークンの返却を要求するトークン返却要求を

送信するトークン返却要求手段として機能させるプログラム。

【請求項１６】

ネットワークによって接続されたストレージに記憶されたファイルの書き込み及び読み出しの操作の権限が、サーバが管理するトークンによって制御されるクライアント装置において実行されるプログラムであって、

前記サーバから送信された、前記ファイルの書込操作権限に関するトークンの返却の要求を受信すると、該ファイルに関するトークンを要求しているクライアント装置に対して、前記記憶部に保持している該トークンに関するファイルを送信する手段として機能させるプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、複数のクライアント（ノード）によってファイルの共有が可能なトークン方式の分散ファイルシステムにおいて、システムのスループットの向上に関するものである。

【０００２】

【従来の技術】

従来の分散ファイルシステムは、トークンを利用してデータのコンシステンシ（一貫性）を保つのもが知られている。このトークンには、データ、属性、サイズ、名前等の種類があり、それぞれのトークンについてリードとライトという状態（ステート）がある。例えば、データリードトークンを保持しているノードは、トークンに対応するファイルへの読み込みが許され、データライトトークンを保持しているノードは、トークンに対応するファイルへの書き込みが許される。また、リードトークンは複数のノードが同時に保持することができるが、ライトトークンはあるひとつのノードのみ保持できる。したがって、同一のファイルは複数のノードが同時に読み込むことができるが、複数のノードが同時に書き込むことができない。

【０００３】

さらにデータトークンに関しては、ファイル全体に対して権限を与えるトーク

ンを発行する方式と、データの中のある部分（例えば、ファイルの先頭何バイト目から何バイト目）に限定した範囲に対する権限を与えるトークンを発行する方式がある。

【０００４】

このトークンは、例えば、図５のようなフォーマットで構成されている。ファイルＩＤフォールド３１は、どのファイルに対してなされたトークンであるかを示す。リボークトークン種フィールド３２は、トークンの種類（データ、属性、サイズ、名前等）を示す。範囲指定フィールド３３は、データの中の範囲を指定する。

【０００５】

この従来のトークン方式の分散ファイルシステムでは、

（１）複数のノードが読み込み専用で同じファイルを参照する、

（２）あるファイルに対してひとつのノードが読み込み、書き込みを行う、

ようなケースでは非常に効率的なファイルアクセスが行える。一方、

（３）複数のノードが同じファイルに対して読み込み、書き込みを行う、

ようなケースでは、パフォーマンスが極端に低下する。

【０００６】

上記（３）のケースの従来の処理の流れを図６のフローチャートを参照して説明する。ここで前提条件として、クライアントＡがファイルＸのライトデータトークンを保持しており、ファイルＸの最新更新データをディスクに未反映（書込前）の状態であるとし、前提条件における処理の流れを太線の矢印で示す。

【０００７】

まず、クライアントＢがファイルＸに対してＩ／Ｏ処理（書込処理又は読出処理）を行った場合（処理１０１）、クライアントＢはファイルＸに対して適切なトークン（リード処理ではリードトークン、ライト処理ではライトトークン）を保持しているか否かを、自身のトークン保持テーブルを参照して判定する（処理１０２）。ファイルＸに対する適切なトークンを保持していると判定された場合は処理１１０に移行する。

【０００８】

一方、クライアントＢはファイルＸに対する適切なトークンを保持していない場合は、サーバＣに対して適切なトークンを要求する。サーバＣは、トークンの要求を受けると、ファイルＸのライトトークンを保持しているクライアント（計算機）があるか否かを、自身のトークン管理テーブルを参照して判定する（処理１０３）。ファイルＸのライトトークンを保持しているクライアントがなければ処理１０８に移行する。一方、クライアントＡがファイルＸに対するライトトークンを保持しているため、サーバＣはクライアントＡに対しトークンリボーク（返却）を要求する。

【０００９】

クライアントＡはサーバＣからのリボーク要求を受けると、自身のキャッシュ内に、ファイルＸのデータであってディスクに未反映の（未だ書き込んでいない）ダーティーデータがあるか否かを判定する（処理１０４）。ダーティーデータがない場合は処理１０６に移行する。一方、クライアントＡにダーティーデータが存在するため、クライアントＡはファイルＸのダーティーデータをディスク（又はサーバ）に書き戻す（処理１０５）。次に、クライアントＡは自身のトークン保持テーブルからファイルＸのライトトークンを削除し、サーバＣに対してトークンの返却を通知する（処理１０６）。サーバＣはトークンの返却通知を受けると、トークンの返却を自身のトークン管理テーブルに反映する（処理１０７）。次に、サーバＣは、クライアントＢに対して要求トークンの付与をトークン管理テーブルに反映し（処理１０８）、クライアントＢに対し要求トークンを付与する旨のメッセージを送る。クライアントＢはメッセージを受け取ると、自身のトークン保持テーブルにトークンの保持を反映する（処理１０９）。クライアントＢのトークンの反映が完了するとディスク（又はサーバ）に対してＩ／Ｏ処理を行ことができる（処理１１０）。

【００１０】

すなわち、あるノードがあるファイルに対してライトデータトークンを保持している状態で、同じファイルに対して別のノードがアクセスしたときに、４回のノード間通信と、２回のディスクＩ／Ｏが発生していた。

【００１１】

【特許文献１】

特開２０００－３２２３０６号公報

【非特許文献１】

日本ＳＧＩ株式会社、“ＣＸＦＳ：クラスタファイルシステム”、［online］、２００１年１１月５日、インターネット＜URL:http://www.sgi.co.jp/products/pdf/sancxfswp.pdf＞

【００１２】

【発明が解決しようとする課題】

従来の分散ファイルシステムでは、あるノードがトークンを保有しているファイルに対して、他のノードがＩ／Ｏ処理をする場合に、２回のディスクＩ／Ｏ処理と４回のノード間通信が発生し、しかもこの処理は順番に実行しなければならないので、ディスクのＩ／Ｏ処理に時間を必要とし、システムの性能が低下していた。

【００１３】

本発明は、上記の不具合を鑑みてなされたものであり、ディスクのＩ／Ｏ処理を減らした分散ファイルシステムを提供することを目的とする。

【００１４】

【課題を解決するための手段】

本発明は、サーバがトークンを保持しているノードに対してリボーク要求を発行する際に、トークンを要求しているノードの情報を付加したリボーク要求を発行することにより、同じファイルをストレージに書き戻さず、ダーティーな状態のままのファイルを渡すので、ストレージへのＩ／Ｏ処理回数を可能な限り減らし、処理を並列化する。

【００１５】

【発明の作用と効果】

本発明は、あるノードがあるファイルに対してライトデータトークンを保持している状態で、同じファイルに対して別のノードがアクセスしたときに、ファイルをストレージに書き戻さず、ダーティーな状態のままのファイルを渡すので、ストレージへのアクセス回数を減らすことができ、理想的なケースでは４回のノ

ード間通信のみ（ディスクＩ／Ｏなし）でファイルへのアクセスが可能となり、システム全体としてのスループットを向上することができる。

【００１６】

例えば、ノード間で共有しているログファイルの処理では、複数のノード間で短いデータの追記書きが発生するため、ディスクのＩ／Ｏ回数を軽減させ、スループット性能の性能向上が期待できる。

【００１７】

【発明の実施の形態】

以下に、本発明の実施の形態を図面を参照して説明する。

【００１８】

図１は、本発明の第１の実施の形態の分散ファイルシステムの構成を表したブロック図である。

【００１９】

クライアント１０、１１は分散ファイルシステムのノードとして機能する計算機である。このクライアント１０、１１とストレージシステム１６とがＬＡＮ１４によって接続されている。

【００２０】

クライアント１０、１１の各々には、トークン保持テーブル１０ｂ、１１ｂ、キャッシュ１０ｃ、１１ｃが備えられている。トークン保持テーブル１０ｂ、１１ｂは、クライアントが保持しているトークンの内容を記憶する。キャッシュ１０ｃ、１１ｃは、ストレージシステム１６から読み込んだデータを一時的に保管するためのメモリ装置である。このキャッシュ１０ｃ、１１ｃには、キャッシュに保持されているデータがダーティーである（ストレージ装置に未だ書き込まれていない未反映のデータがある）か否かを示すダーティーフラグが備えられている。

【００２１】

ストレージシステム１６には、サーバ１２、ストレージ装置１３が備えられており、Network Attached Storage（ＮＡＳ）として機能している。サーバ１２とストレージ装置１３とはStorage Area Network（ＳＡＮ）１５によって接続され

ている。このストレージ装置１３は、ハードディスク等の記憶装置で構成されている。よって、クライアント１０、１１は、ＬＡＮ１４を経由してストレージシステム１６に備えられたストレージ装置１３に記憶されたデータの読み出し及び書き込みができる。

【００２２】

サーバ１２には、トークン管理テーブル１２ａ、トークン保持テーブル１２ｂ、キャッシュ１２ｃが備えられている。トークン管理テーブル１２ａは、全てのクライアントが、それぞれどのような内容（例えばリード、ライト等）のトークンを保持しているかを該当するデータに対応付けて記憶する。トークン保持テーブル１２ｂは、サーバが保持しているトークンの内容を記録する。キャッシュ１２ｃは、データを一時的に保管するためのメモリである。

【００２３】

次に、以上のように構成された本発明の第１の実施の形態の動作について説明する。

【００２４】

各クライアント１０、１１はサーバ１２からトークンを付与されることによってファイルへのアクセスが許され、トークンを管理しているサーバ１２は、トークン管理テーブル１２ｃを参照することによって、現在、どのクライアントによって、どのファイルに対してどのような処理が行われているのかを知ることができる。

【００２５】

例えば、クライアント１０にファイルＡに対するライトデータトークンが付与されると、クライアント１０は、ストレージ装置１３からファイルＡを読み込んだり、ファイルＡを書き込みモードでキャッシュ１０ｃに保持することが可能になる。続いて、クライアント１１がファイルＡに対するリードデータトークンを要求した場合、サーバ１２が、クライアント１１からのリードデータトークン要求を受け取ると、トークン管理テーブル１２ｃを参照することで、ファイルＡがクライアント１０が保有するライトデータトークンによって書き込みモードで保持されていることがわかる。そのため、サーバ１２はクライアント１０に対して

トークンのリボーク（返却）要求を行う。リボーク処理が完了すると、クライアント１１にファイルＡに対するリードデータトークンが付与され、クライアント１１によるファイルＡの読み込みが可能となる。

【００２６】

図２は、第１の実施の形態のクライアント１０、１１とサーバ１２のファイルのＩ／Ｏ処理（ファイルの読み込み、書き込み等）のフローチャートを示す。

【００２７】

なお、このフローチャートの処理の前提条件として、クライアント１０がファイルＸに対するライトデータトークンを保持しており、ファイルＸの最新更新データがディスクに未反映の状態、すなわちファイルＸにダーティーデータを含んでいる状態を前提し、前提条件における処理の流れを太線の矢印で示す。

【００２８】

まず、クライアント１１において、ファイルＸを読み込む必要が生じ、分散ファイルシステム上のファイルＸに対するＩ／Ｏ要求を発行する（処理２０１）。

【００２９】

クライアント１１は、このＩ／Ｏ要求を受けて、まず、自身のトークン保持テーブル１１ｂがファイルＸの適切なトークンを保持しているか否かを判定する（処理２０２）。適切なトークンを保持している、すなわち、ファイルＸに対する読み込み処理ならばリードデータトークン、ファイルに対する書き込み処理ならばライトデータトークンを保持していると判定した場合には、サーバ１２に対してＩ／Ｏ処理を実行し、ストレージ装置１３よりファイルを読み込む（処理２１８）。

【００３０】

処理２０２において適切なトークンを保持していないと判定した場合には、サーバ１２に対してトークンを要求するトークン要求メッセージを送る。サーバ１２はトークン要求メッセージを受け取ると、ファイルＸのライトデータトークンを保持するクライアントがトークン管理テーブル１２ａに登録されているか否かを判定する（処理２０３）。ライトデータトークンを保持しているクライアントが無い場合は、直ちにトークンを付与することが可能であるので、トークン管理

テーブル１２ａにファイルＸに対するクライアント１１のライトデータトークンの付与を反映し（処理２１６）、クライアント１１にライトデータトークンを付与した旨を通知する。

【００３１】

クライアント１１は通知を受け取ると、トークン保持テーブル１１ｂに、処理２０２で要求したトークンの種類（ライト／リード等）を登録し（処理２１７）、サーバ１２に対してＩ／Ｏ処理を実行し、ストレージ装置１３よりファイルを読み込む（処理２１８）。

【００３２】

処理２０３において、ライトデータトークンを保持しているクライアントが既に存在すると判定した場合には、ライトデータトークンを保持しているクライアント（クライアント１０）に対してリボーク要求メッセージを送る（処理２０４）。このリボーク要求メッセージには、図３にて後述するように、トークンを要求しているノード（クライアント１１）、要求しているトークンのステート（クライアント１１が要求しているステート（リード又はライト））の情報が含まれている。

【００３３】

クライアント１０は、リボーク要求メッセージを受け取ると、ライトデータトークンのリボーク処理（ライトデータトークンの返却とファイルの受け渡し）を行う。まず、トークン保持テーブル１０ｂからファイルＸに対するライトデータトークンを削除する（処理２０５）。次に、クライアント１０のキャッシュ１０ｃにあるファイルＸのデータがサーバ１２に反映されているかどうか、すなわちファイルＸにダーティーデータが存在するか否かを判定する（処理２０６）。ダーティーデータが存在すると判定した場合には、ダーティーデータを含んだファイルＸをトークン要求元（クライアント１１）に対して送信する（処理２０７）。このトークン要求元はトークン要求メッセージに含まれているトークン要求ノードより得る。ファイルＸを送ると処理２０８に移行する。

【００３４】

一方、処理２０６においてダーティーデータが存在しないと判定した場合には

、処理２０８に移行する。

【００３５】

次に、ファイルＸを受け取ったトークン要求元（クライアント１１）は、受信したデータを自身のキャッシュ１１ｃに取り込む（処理２１１）。このとき、クライアント１１が処理２０２において要求したトークンの種類がリードデータトークンである場合には（処理２１２）、直ちにファイルＸをサーバ１２に書き出す処理（ファイルの同期処理）を行う（処理２１３）。これは、リードデータトークンのみを保有しているクライアントがダーティーデータを含むファイルを持つと、ファイルの一貫性（コンシステンシ）が失われるためである。すなわち、複数のクライアントが同時にファイルを読み出すことができるので、ディスクに未反映のデータがあるとファイルの一貫性が保てないためである。

【００３６】

このとき、クライアント１０は、処理２０５で受け取ったリボーク要求メッセージからトークン要求元の要求トークンステートフィールドを参照して、要求トークンの種類（要求トークンがリードであるかライトであるか）を判定する（処理２０８）。トークン要求元がライトデータトークンを要求していると判定した場合には、自身の持つキャッシュ１０ｃにあるファイルＸのダーティーデータを全て破棄する（処理２０９）。これは、同一のファイルに対して複数のクライアントにライトトークンを付与することはできないため、他のノードがライトデータトークンを要求した場合には、リボーク処理によってファイルの所有権を新たにライトデータトークンを保有するクライアントに完全に移行するためである。処理が完了すると、トークン要求元（クライアント１１）に対して自身のトークンステートの変化を通知する。

【００３７】

一方、処理２０８でトークン要求元がリードデータトークンを要求していると判定した場合には、キャッシュ内にあるファイルＸのダーティーデータに対応するダーティー状態を示すダーティーフラグをクリアし、データ自体は破棄せず読み込み専用データに変更してキャッシュ１０ｃに保存しておく。次に、自身のトークン保持テーブルにファイルＸのリードデータトークンを登録する（処理２１

０）。これは同一のファイルに対してリードデータトークンは複数のクライアントが持つことができるためであり、ファイルＸの中のダーティーデータは処理２１３によってディスクに反映されるからである。処理が完了すると、トークン要求元（クライアント１１）に対して自身のトークンステートの変化を通知する。

【００３８】

ここで、クライアント１１は、処理２０９又は処理２１０によって、クライアント１０からトークンステートの変化の通知を受け、さらに処理２１３が完了すると、クライアント１０（トークンリボーク要求先）と自身（トークン要求元であるクライアント１１）のトークンステートの変化をサーバ１２に送信する（処理２１４）。なお、このクライアント１０のトークンステートの変化を、クライアント１１を経由することなく、クライアント１０からサーバ１２に直接送信するようにしてもよい。

【００３９】

サーバ１２はこの変化をトークン管理テーブル１２ａに反映する（処理２１５）。

【００４０】

次に、クライアント１１は、自身のトークン保持テーブル１１ｃに、処理２０２で要求したトークンの登録を行う（処理２１７）。そして、必要ならばサーバ１２に対してＩ／Ｏ処理を実行し、ストレージ装置１３よりファイルを読み込む（処理２１８）。

【００４１】

以上の処理によって、クライアント１０に対するトークンのリボーク処理が完了し、ダーティーデータを含んだデータＸがクライアント１０からクライアント１１へ渡される。

【００４２】

なお、上記の処理では、クライアント１０からクライアント１１に対してダーティーデータを含んだファイルＸの送信（処理２０７→処理２１１）とトークン状態変化通知の送信（処理２０９、２１０→処理２１４）は別に行っているが、ひとつのメッセージにまとめて一度に送信することも可能である。

【００４３】

また、図２の処理２０７おいて、クライアント１１に送る代わりにストレージ装置１３に書き出した方が性能上有利である場合、すなわちクライアント（またはサーバ１２）が、ＬＡＮのスループット性能やストレージ装置のＩ／Ｏのスループット性能等から予め設定しておいた性能予測式に基づいて、ディスクのＩ／Ｏスループット性能がＬＡＮのスループット性能より高いと判断した場合には、要求クライアントに対してファイルを送らずに、ストレージ装置１３に書き出す処理を行いダーティー状態でなくなったファイルを、要求クライアントが読み出すようにしてもよい。こうすることで、スループットの高い処理手段を優先でき、分断ファイルシステム全体としてのスループットが向上する。

【００４４】

図３に、図２に処理２０４において送信されるトークン要求メッセージの詳細を示す。

【００４５】

トークン要求メッセージは、ファイルＩＤフィールド４１、リボークトークン種フィールド４２、範囲指定フィールド４３、要求トークンステートフィールド４４、トークン要求ノードＩＤフィールド４５から構成される。

【００４６】

ファイルＩＤフィールド４１は、トークンの要求を行うファイルの情報を示している。リボークトークン種フィールド４２は、トークンの種別（例えばデータ、属性、サイズ、ファイル名等）を示す。範囲指定フィールド４３は、ファイルの範囲を示す。これはトークンによってファイルの範囲指定が可能な分散ファイルシステムにおいて、この範囲指定フィールド４３で指定した範囲内のみでクライアントが処理を行うことができるものである。要求トークンステートフィールド４４は、トークンのステート（リード／ライト）を示す。トークン要求ノードＩＤフィールド４５は、トークンを要求している要求元のクライアントの情報を示す。

【００４７】

以上のように構成された本発明の第１の実施の形態の分散ファイルシステムで

は、あるクライアントが持っているダーティーデータを含んだファイルを他のクライアントが要求した場合に、ファイルをストレージに書き戻さず、ダーティーな状態のままのファイルを他のクライアントに渡すので、ストレージへのアクセス回数を減らすことができ、分散ファイルシステム全体のスループットを向上することができる。

【００４８】

次に、本発明の第２の実施の形態の分散ファイルシステムについて説明する。第２の実施の形態では、第１の実施の形態と比較すると、各クライアントが直接ストレージとＳＡＮによって接続されており、サーバを経由することなくストレージ上のデータを読み書きできる点が相違する。なお、第１の実施例と同一の動作をする構成には同一の符号を付し、その説明は省略する。

【００４９】

図４は第２の実施の形態の分散ファイルシステムの構成を表したブロック図である。

【００５０】

クライアント１０、１１とストレージシステム２６（サーバ２２）とがＬＡＮ２４によって接続されている。

【００５１】

ストレージシステム２６には、サーバ２２、ストレージ装置１３が備えられており、サーバ２２とストレージ装置１３とはＳＡＮ２５によって接続されている。

【００５２】

また、クライアント１０、１１も、ストレージ装置１３とＳＡＮ２５によって接続されている。そのため、クライアント１０、１１はサーバ２２を介することなくストレージ装置１３にあるファイルを読み込み、ストレージ装置２３に対してファイルを書き込むことができる。すなわち、クライアント１０、１１、サーバ２２、ストレージ装置１３がＳＡＮを構成している。

【００５３】

サーバ２２には、トークン管理テーブル２２ａ、トークン保持テーブル２２ｂ

、キャッシュ２２ｃが備えられている。トークン管理テーブル２２ａは、各クライアントから発行されたトークンを受け取り、受け取ったトークンは、どのクライアントから、どのような内容（例えばリード、ライト等）のトークンであるかを該当するデータに対応付けて記憶する。トークン保持テーブル２２ｂは、クライアントが要求したトークンの内容を保持する。キャッシュ２２ｃは、データを一時的に保管するためのメモリである。

【００５４】

次に、上記のように構成された第２の実施の形態の分散ファイルシステムについて、次に動作を説明する。

【００５５】

本実施の形態の処理の流れは、第１の実施の形態（図２）と基本的には同一であるが、クライアント１０、１１は、ストレージ装置１３に対するＩ／Ｏをサーバを介することなく行うことができる。図２の処理２１３では、ファイルＸをサーバ１２に対して送った後にストレージ装置１３に書き込まれるが、第２の実施の形態では、クライアント１１が直接ストレージ装置１３に対してＩ／Ｏを行いファイルＸを書き込む。同様に、処理２１８では、クライアント１１が直接ストレージ装置１３に対してファイルＸに対するＩ／Ｏを発行する。

【００５６】

以上のように構成された第２の実施の形態では、第１の実施の形態の効果に加え、クライアントがサーバ１２を介すことなくストレージ装置にＩ／Ｏが行えるので、サーバ１２の処理が軽減され、結果として分散ストレージシステム全体としてのスループットを向上することができる。

【図面の簡単な説明】

【図１】　本発明の第１の実施の形態の分散ファイルシステムの構成を示すブロック図である。

【図２】　同じく分散ファイルシステムの処理を示すフローチャートである。

【図３】　同じくリボーク要求トークンの内容を示すフォーマット図である。

【図４】　本発明の第２の実施形態の分散ファイルのシステムの構成を示すブロック図である。

【図５】　従来のトークンリボーク要求の内容を示すフォーマット図である。

【図６】　従来の分散ファイルシステムの処理を示すフローチャートである。

【符号の説明】

１０、１１　クライアント

１０ｂ、１１ｂ　トークン保持テーブル

１０ｃ、１１ｃ　キャッシュ

１２　サーバ

１２ａ　トークン管理テーブル

１２ｂ　トークン保持テーブル

１２ｃ　キャッシュ

１３　ストレージ装置

１４　ＬＡＮ

１５　ＳＡＮ

１６　ストレージシステム

２０　クライアント

２２　サーバ

２２ａ　トークン管理テーブル

２２ｂ　トークン保持テーブル

２２ｃ　キャッシュ

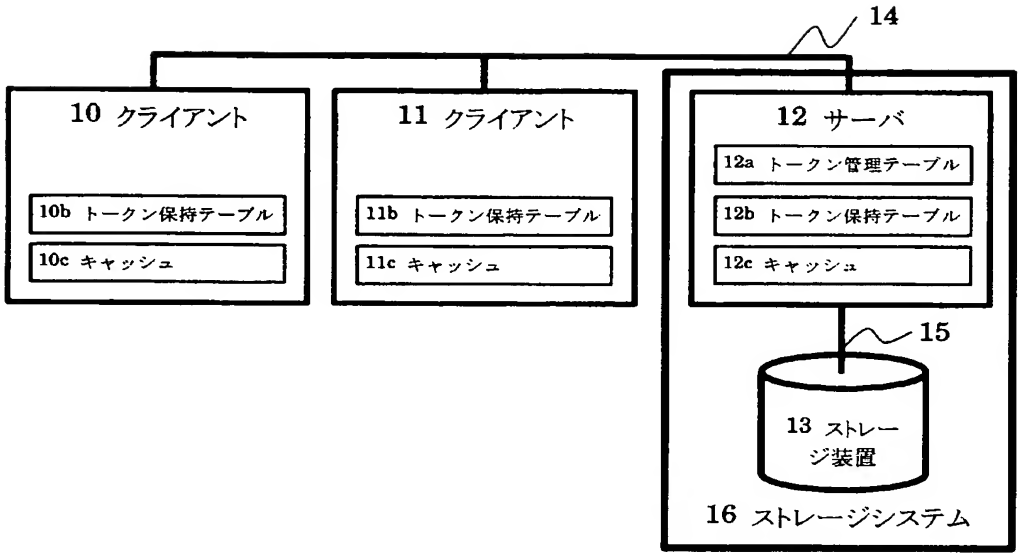
２３　ストレージ装置

２５　ＳＡＮ

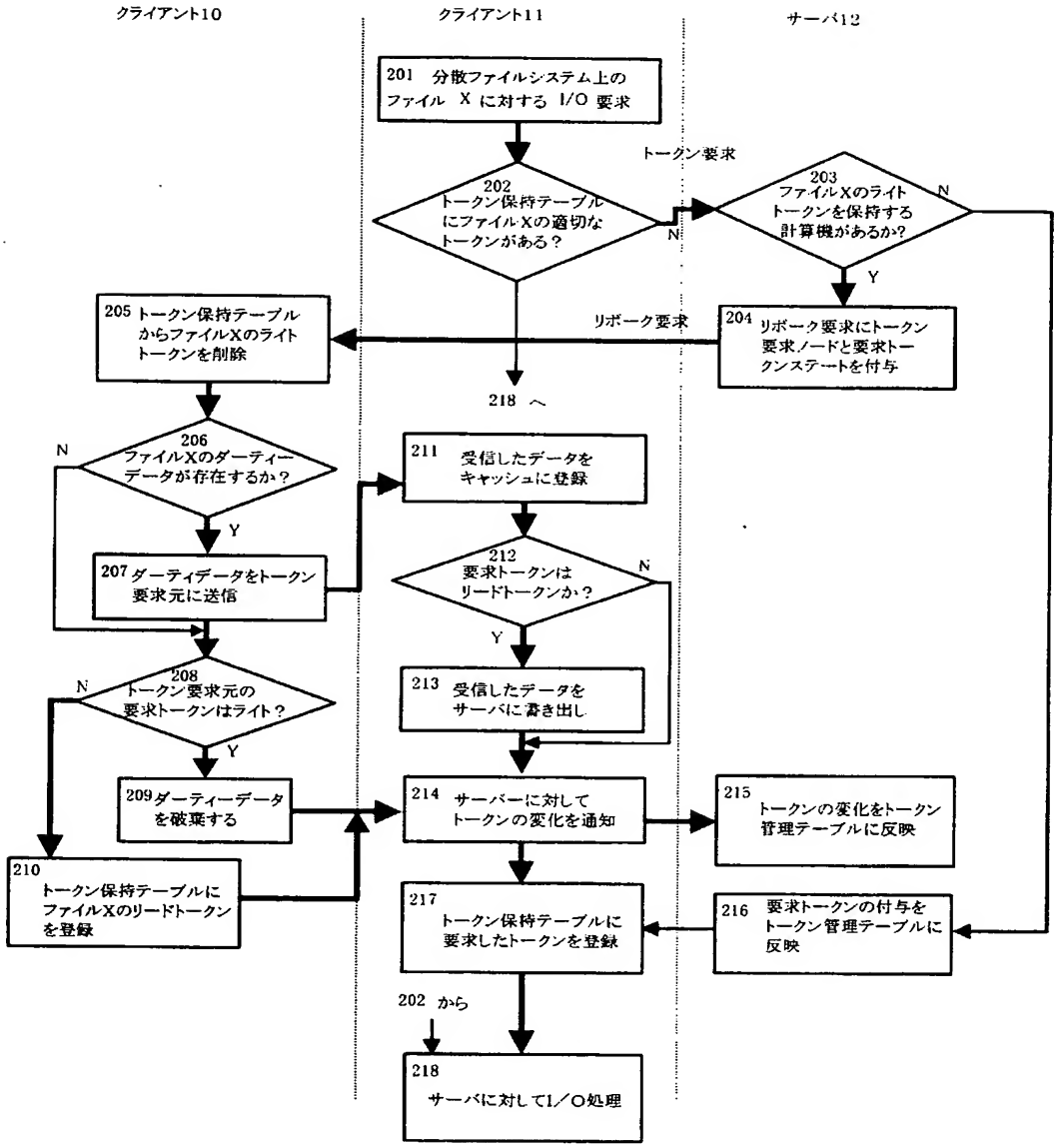
２６　ストレージシステム

【書類名】　図面

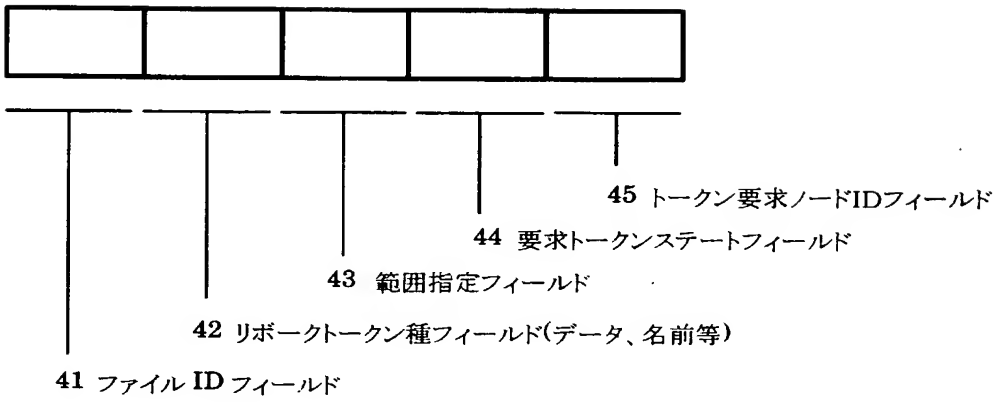
【図１】


14
10 クライアント
10b トークン保持テーブル
10c キャッシュ
11 クライアント
11b トークン保持テーブル
11c キャッシュ
12 サーバ
12a トークン管理テーブル
12b トークン保持テーブル
12c キャッシュ
15
13 ストレージ装置
16 ストレージシステム

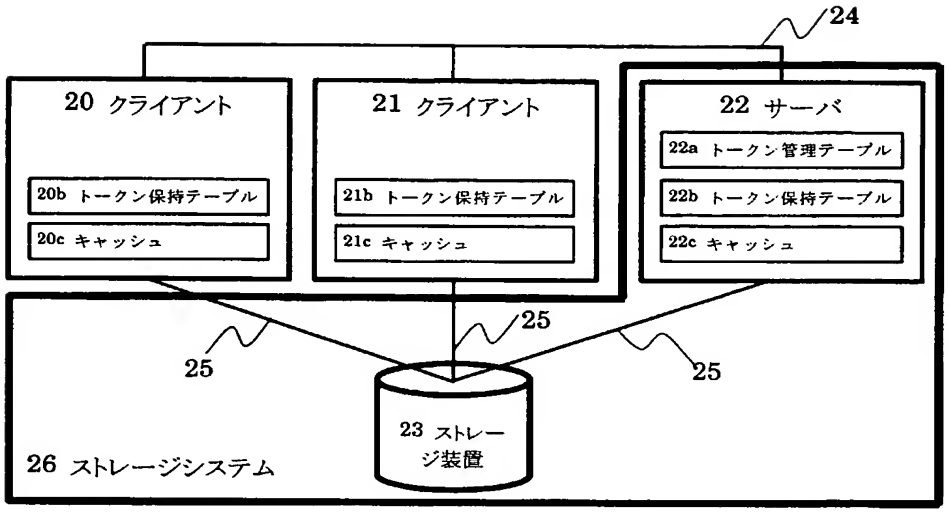
【図２】


クライアント10
クライアント11
サーバ12
201 分散ファイルシステム上のファイル X に対する I/O 要求
202 トークン保持テーブルにファイルXの適切なトークンがある？
トークン要求
203 ファイルXのライトトークンを保持する計算機があるか?
N
Y
リボーク要求
204 リボーク要求にトークン要求ノードと要求トークンステートを付与
205 トークン保持テーブルからファイルXのライトトークンを削除
218 へ
206 ファイルXのダーティーデータが存在するか？
N
Y
207 ダーティデータをトークン要求元に送信
208 トークン要求元の要求トークンはライト？
N
Y
209 ダーティーデータを破棄する
210 トークン保持テーブルにファイルXのリードトークンを登録
211 受信したデータをキャッシュに登録
212 要求トークンはリードトークンか？
N
Y
213 受信したデータをサーバに書き出し
214 サーバーに対してトークンの変化を通知
215 トークンの変化をトークン管理テーブルに反映
216 要求トークンの付与をトークン管理テーブルに反映
217 トークン保持テーブルに要求したトークンを登録
202 から
218 サーバに対してI／O処理

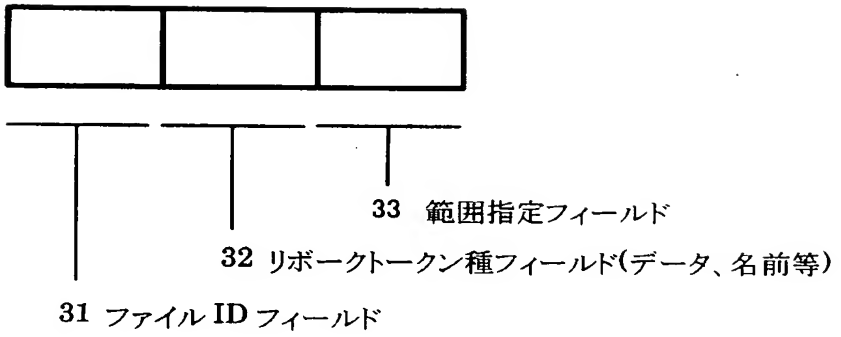
【図３】


45 トークン要求ノードIDフィールド
44 要求トークンステートフィールド
43 範囲指定フィールド
42 リボークトークン種フィールド(データ、名前等)
41 ファイル ID フィールド

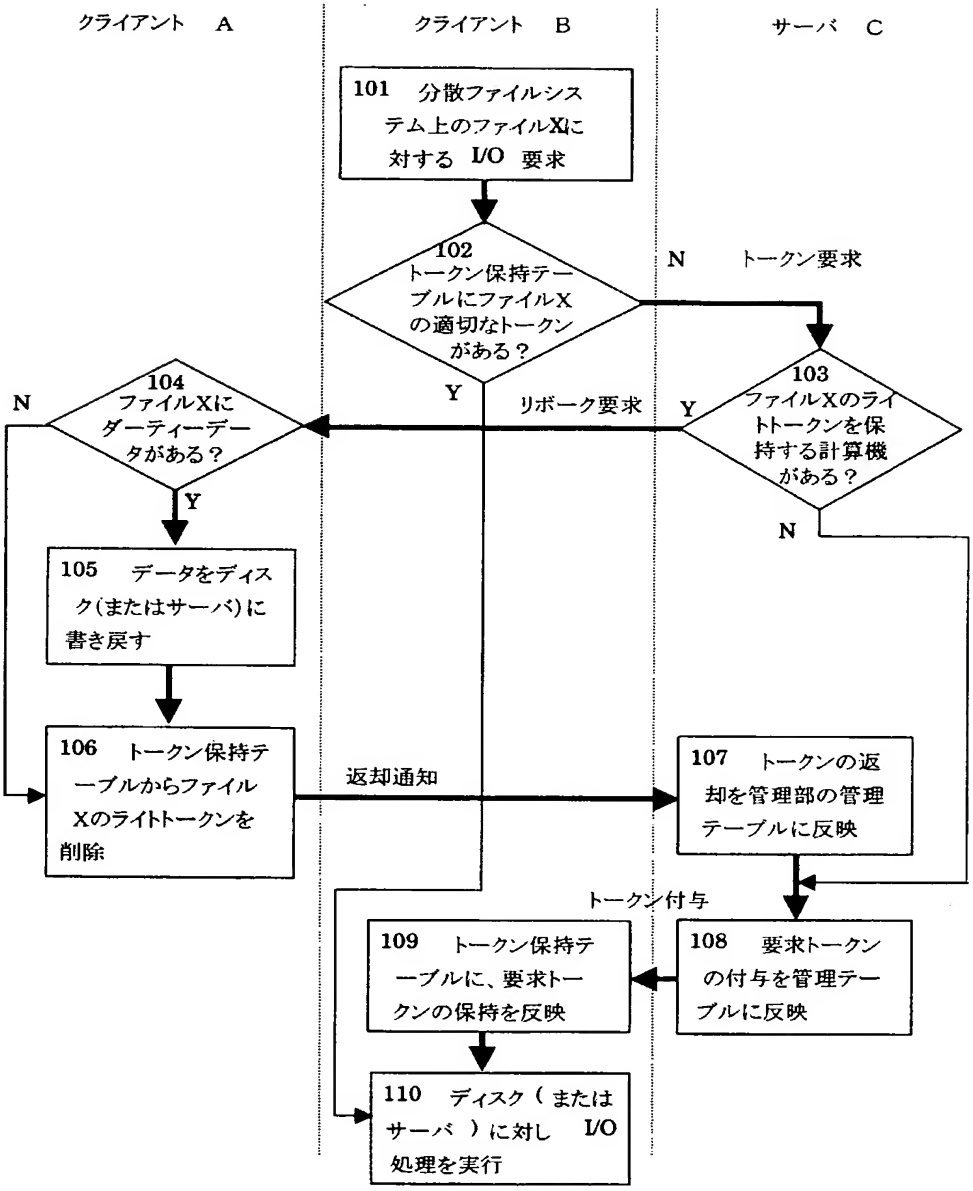
【図４】


24
20 クライアント
20b トークン保持テーブル
20c キャッシュ
21 クライアント
21b トークン保持テーブル
21c キャッシュ
22 サーバ
22a トークン管理テーブル
22b トークン保持テーブル
22c キャッシュ
25
25
25
23 ストレージ装置
26 ストレージシステム

【図５】



【図６】


クライアント　A
クライアント　B
サーバ　C
101　分散ファイルシステム上のファイルXに対する I/O 要求
102 トークン保持テーブルにファイルXの適切なトークンがある？
N
トークン要求
Y
リボーク要求
103 ファイルXのライトトークンを保持する計算機がある？
Y
N
104 ファイルXにダーティーデータがある？
N
Y
105　データをディスク(またはサーバ)に書き戻す
106　トークン保持テーブルからファイルXのライトトークンを削除
返却通知
107　トークンの返却を管理部の管理テーブルに反映
トークン付与
108　要求トークンの付与を管理テーブルに反映
109　トークン保持テーブルに、要求トークンの保持を反映
110　ディスク（またはサーバ）に対し I/O 処理を実行

【書類名】　要約書

【要約】

【課題】

複数のクライアント（ノード）によってファイルの共有が可能なトークン方式の分散ファイルシステムにおいて、システムのスループットの向上に関する

【解決手段】

ファイルを保持するストレージ１３と、トークンの管理を行うサーバ１２と、ストレージ１３に対してファイルの読み書きを行う複数のクライアント１０と、がネットワークで接続されており、前記クライアントがトークンによってファイルの要求を行う分散ファイルシステムにおいて、サーバ１２は、ファイルを保持しているクライアント１０に対してトークンの返却を要求するトークン返却要求手段を備え、前記トークン返却要求手段は、トークンによってファイルを要求しているクライアント１０の情報と、トークンの内容を示す情報とを含めたトークン返却要求を送信する。

【選択図】　図１

特願２００３－０６３５６９

出　願　人　履　歴　情　報

識別番号　　［０００００５１０８］

１．変更年月日　　１９９０年　８月３１日
［変更理由］　　新規登録
住　所　　東京都千代田区神田駿河台４丁目６番地
氏　名　　株式会社日立製作所